12 SSD Assessment Task 1

**Question 1**

1. Classify each of the languages used in terms of the computer language generation to which it belongs and justify your answers.

* **C**: The programming language C, is a powerful but rather cryptic language capable of precisely specifying computer operations categorized as a 3rd generation language, also known as “high-level”. The language C is a programming language that has facilities for structured programming and allows lexical variable scope and recursion, while a strict type system prevents many unintended operations. Not to mention, the language is compatible, highly efficient and its general-purpose programming language is versatile.
* **Forth**: Forth, a concise but versatile procedural language originally designed to regulate movement of telescopes and also used to control devices and processes ranging from heart monitors to special-effects video cameras. This specific programming language was created for programming embedded and real-time applications, allowing it to cross compile, generating highly optimized code that runs on a variety of microprocessors and micro controllers, and prove themselves very capable in custom-hardware environments. The forth programming langue is a high-level programming language, although most versions include an assembler.
* **Assembly language**: Assembly language is a low-level programming language designed for microprocess and other programmable devices. The language may be produced by compiling source code form a high-level programming language. Each assembly language is specific to a computer architecture. In contrast, most high-level programming languages are generally portable across multiple architectures but require interpreting or compiling Assemble code can be converted to machine code using an assembler. An example of assembly language would be programming the telemetry system at either end of the finger thick coax cable connecting the vessels, which in effect enabled their computers to talk to each other.

1. Why is assembly language described here as rudimentary (Elementary or undeveloped) and in this case why would it be used?

* Programming language assembly is considered rudimentary in this situation considering the high-level language. However, is still the fastest and most efficient language you could use, because you have precise control over every clock cycle and byte. Conversely, runs five times faster than code produced by an optimised compiler, in addition it can sometimes run an order of magnitude faster. The reason it was used in this situation was due to its primary purpose of direct hardware manipulation, access to specialized processor instructions. Typically, uses are devices drivers and real-time systems. Though the advantages are true, the chance or programming an application in this program has a low probability.

1. Both fourth and C are imperative languages. Compare and contrast imperative and object-oriented languages.

* Imperative programming is a programming language paradigm that uses statements that change a programs state. An imperative program consisted of a set of commands for the computer to execute. The imperative programming focuses on describing how a program operates. The term is used mainly when comparing to declarative programming, in which focuses on what the program should accomplish without specifying how the program should achieve the result. In comparison, Object oriented programming is entirely based on the concepts of such called “objects”. These objects may contain data, in the form of fields often known as attributes and methods. A feature is that an object can access and often modify the data fields of the object with which they’re associated with. Programmes including C++, java and python are multi paradigm programming language that support the use of object oriented programming in combination with imperative programming.

1. Explain the use of three different languages in this scenario rather than a single all-purpose language.

* The use of 3 languages were for different specific tasks, such as C for programming computer operations, the forth language to regulate movement of telescopes and used to control devices and processing ranging from heart monitor and assembly language for its reliability of being able to alter clock speeds for process and general programming of the process. All in all, different programming language for different tasks.

**Question 2**

1. Describe all possible results from running the code on line 22.

Line 22: go:-parent(X, Melody)

* The rule on line 22 (go:-parent(x, melody)) output the results Jim and Eleanor. This is shown through the rule that determines whether they are the parents. Line 19 (. parent(X, Y) :- father (X, Y).)) Is set to determine the child’s father and not to mention it’s the same for the mother (parent(X, Y) :- mother (X, Y).).

1. father(micheal, cathy).

father(micheal, sharon).

father(charles\_gordan, micheal).

father(charles\_gordan, julie).

father(jim, melody).

father(jim, crystal).

father(elmo, jim).

father(greg, stephanie).

father(greg, danielle).

mother(melody, cathy).

mother(melody, sharon).

mother(hazel, michael).

mother(hazel, julie).

mother(eleanor, melody).

mother(eleanor, crystal).

mother(crystal, stephanie).

mother(crystal, danielle).

sibling(X, Y) :- mother(M, X), father(F, X), mother(M, Y), father(F, Y), X \= Y.

**Question 3**

A school is developing a new administration program using Object Oriented Programming (OOP). It will need to store data about teachers, office staff and students. The data to be stored include data-of-birth, address and home phone number. In addition, extension phone number and pat details will be stored for teachers and office staff, and year level will be stored for students.

A class called person is to be developed, describe an attribute and a method for this class, using examples.

Class person

{

private:

Std::string address;

public:

std::string getaddress()

(

return address();

)

);

* The class ‘person’ includes an attribute “Std::string address;” and “return address();” for the method.

**Question 4**



Link to the program: <https://repl.it/@dabdabdab/GenerousSecretQueenslandheeler>

**Question 5**

This C++ program consist of demonstrating the following:

Abstraction:

Abstraction is the process that refers to providing essential information such as characteristics to the outside world and hiding their background details such as making it private. Data abstraction is a programming technique that relies on the separation of interface and implementation. In C++, classes provide a greater level of abstraction, such as proving sufficient public methods to the outside world to play with the functionality of the object and to manipulate object data, i.e., state without actually knowing how class has been implemented internally.

Example: class Person //This class contains data abstraction, having the basic concepts and features.

{

private: // This section is my revealing of encapsulation. Accessible only to member function.

int age;

int phonenumber;

bool hsc;

public:// This section is accessible anywhere in the code.

int getAge()

{

return age;

}

void setAge(int a)// Getters and Setters.

{

age = a;

The members are declared as public and public in a class.” public:// This section is accessable anywhere in the code.” and private: // This section is my revealing of encapsulation. Accessable only to member function.

Inheritance:

Inheritance is the capacity of a class to derive properties and characteristics form another class. This is one of the most important features in Object Oriented programming, providing an opportunity of reusing the code functionally and fast. When creating a class, instead of writing complete new data members and members functions the programmer can designate that the new class should inherit the members of the existing class.

Example: class Student: public Person // This is inheritance of person.

{

Person person;

public:

int year\_group() //polymorphism of the attributes from the first class overriding it.

{

y = person.getAge();

return y;

}

This is the exmaple of inheritance shown in my program. “class Student: public Person”

Polymorphism:

Polymorphism substantiates the forms of which it may take. This means polymorphism is defined as the ability to be displayed in more than one or more forms. Polymorphism means that a call to a member function will cause a different function to be executed depending on the specific type of object that provokes this function to execute. In C++ polymorphism is mainly divided into two types: the Compiler time and Runtime. Compile time polymorphism is achieved by overloading, when there are multiple functions with the same name but different parameters. Runtime error is the function overriding which occurs when derived from a class that has a different definition for the one of the member’s functions of the base class.

Example: int year\_group() //polymorphism of the attributes from the first class overriding it.

Polymorphism is overring the attribute from the superclass.

Encapsulation:

The term encapsulation is defined as combining data members and functions in a single unit called a class. In object oriented programming, encapsulation is the concealing of data. Which is then used to bind together the data and functions that manipulate the data, and that keeps both safe from outside.

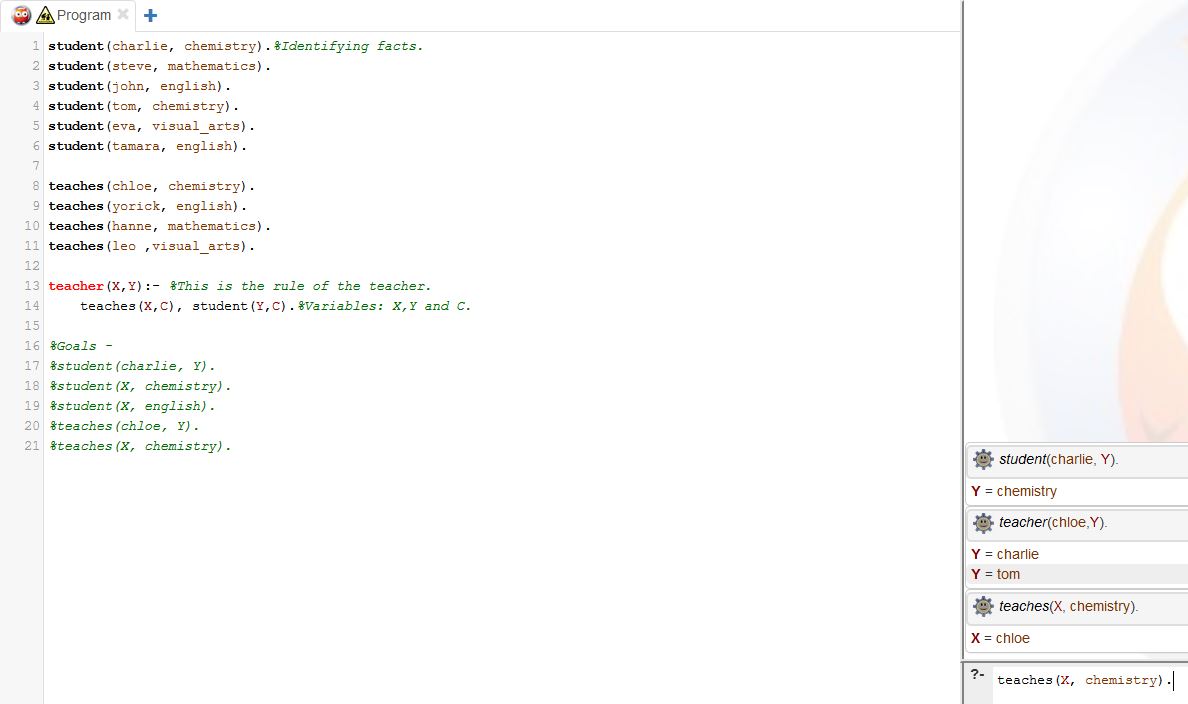
Example: private: // This section is my revealing of encapsulation. Accessible only to member function.

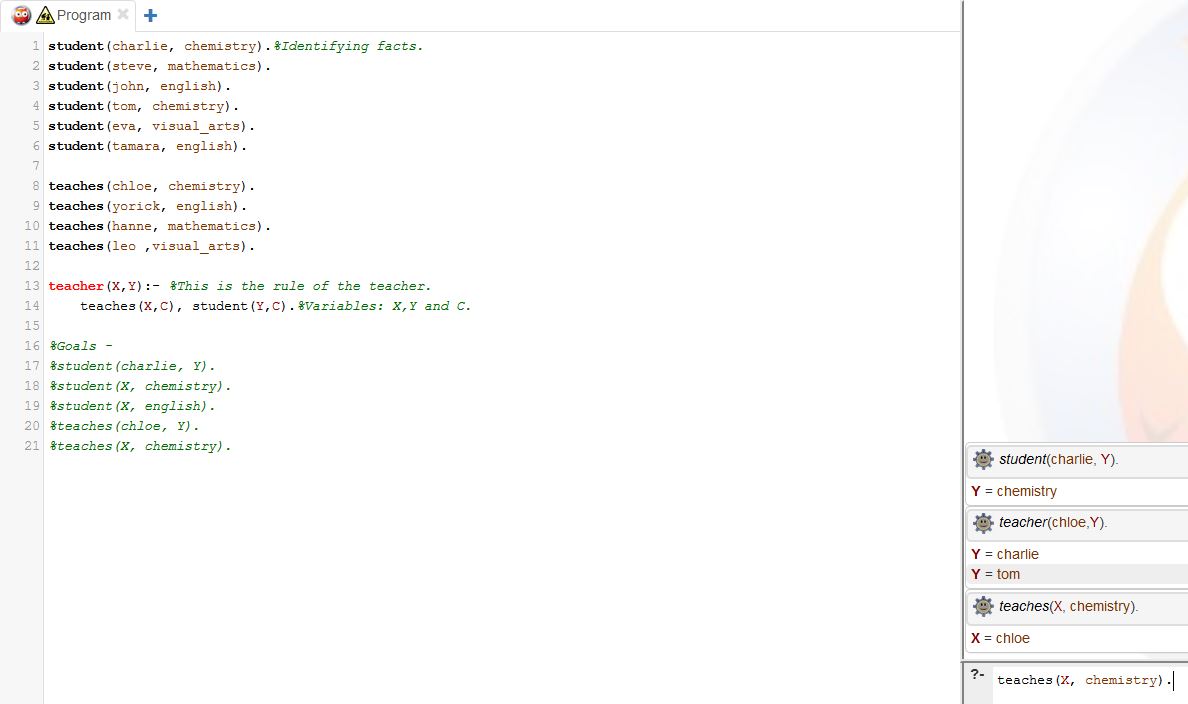
int age;

int phonenumber;

bool hsc;

**Question 6**





Link to the program: <https://swish.swi-prolog.org/p/Teacher%20and%20Student.pl>

**Question 7**

The prolog program demonstrates the use of the following:

Heuristics: A rule of thumb often based on experience. It can results in one or more possible solutions. This is shown as the end results such as ‘teacher(chloe, Y).’ This outputs Y = Charlie and Y = Tom.

Inference engine: The control mechanism that applies knowledge, contained in knowledge base to resolve goals. The corresponding demonstration of Inference engine is shown during the stage of inputting something into the query such as: ‘teaches(chloe, chemistry)’ this comes up as true or ‘teachers(hanne, english)’ as false.

Backward/Forward Chaining: Forward chaining is the start of the knowledge base and apply the rules and facts to arrive at a conclusion. In comparison, backwards chaining is starting with the goal then works backwards to prove the validity of each statement.

**Question 8**

This is a peer review of sarthak’s program. The program contains a concise but generic interface and structural integrity is needed to improve the impairments. Sarthak, could add a “Welcome to (game)” when initiating the code, not to mention, the code could also implement a selection structure to choose what attacks, specials and weapons for his avatar. Sarthak’s program runs perfectly fine and smoothly without any errors. However, his program could use more comments regarding his demonstration of polymorphism, inheritance, data abstraction and encapsulation. Besides, that sarthak has structure his program they well, his program doesn’t have any errors, code instantiations are well done similarly to his methods, and demonstrates all the necessary demonstrations of encapsulation, polymorphism, data abstraction and inheritance.